## Introduction

Individual elements in complex systems can affect one another. Consider an ecosystem in a forest. Think of each organism in the forest as a node that has a graph edge to every other organism that it comes in contact with. Because this is a fairly large forest, many of the organisms in the forest will never come into contact with each other, and thus do not require an edge. This is the idea of *sparsity* in a graph, only some possible edges are actual edges.

Each edge in this graph is a directed edge with some real numbered value. Positive values are for beneficial relationships, i.e., a tree has a positive relationship to a bird because the tree provides shelter, or a rabbit has a positive relationship to a fox because the rabbit is food. Conversely, the fox would have a strongly negative edge to the rabbit, indicating a detrimental relationship. This network models the functional relationships between the nodes, and gives us a picture of the interactions between organisms in the forest.

Looking at how systems change over time can give valuable insights into how each member of the system relates to other members. [Jesse: make this active and add in examples: Knowing the interactions between members of a system can lead to inferences. For example, more foxes may lead to fewer rabbits in the short term, but then fewer rabbits will lead to fewer foxes in the medium term.] By learning the interactions between members of a system, new aspects of their functionality can be inferred. And once new functionality is inferred, this can drive future experiments. Network inference can thus be looked at both as a hypothesis generating and an explanatory framework.

\*\*\* Some more introduction and cool things about network inference \*\*\* They are visual. The quantification of edges can allow us to filter out less important edges in favor of more important ones.

## Definition

The motivation behind building gene regulatory networks is similar to building ecosystem edges in the forest example above. Instead of organisms, each node is now a gene, and the edges between genes indicate some sort of relationship. For example, if whenever Gene 1’s expression value goes up, Gene 2’s expression value also goes up, then we place … [Jesse: active voice] A positively weighted directional edge can be placed from Gene 1 to Gene 2. Adding a third gene, Gene 3, whose expression value is reduced when Gene 2’s goes up, allows the beginning of a formation of an interesting network. From this extremely simple network, it can be inferred that if something causes Gene 1’s expression value to increase, Gene 3’s will decrease sometime later.

A gene regulatory network (GRN) [Jesse: no acronyms except DNA] is a graph where the nodes represent genes, and the edges between the nodes show how different genes affect one another. Gene regulatory networks can answer questions such as “When Gene 1 becomes expressed, what happens to Gene 2?”. Gene regulatory networks give insight into the function and organization of how genes work in an organism.

[Jesse: I think you don’t need to say any more about importance. The next question is how to do the inference.]

Relationships between genes can be inferred by looking for dependencies and trends in gene expression data. For example, if every time Gene 1 became expressed, Gene 2 became expressed shortly thereafter, a dependency may be inferred. These inferred edges can be used to construct a network that represents interactions between individual genes and between groups of genes.

## Why Important

Understanding how genes regulate each other is important in the explanation of functional relationships. For example, while it may be known that a given gene codes for a specific protein, understanding the regulatory network around that gene allows understanding of *how* that gene becomes activated or repressed, and what happens to other genes down the line.

GRNs can also be used to infer new functionality of genes. Inferring the connections between nodes in the network allows for learning of new functionality and interactions. Different behavior of genes may be observed when looking at time-series data versus steady state. On the system’s way to an equilibrium state, genes may function in different groups, sub-networks can interact with each other, or other possibly novel interactions may occur that would be missed if only steady-state data were observed.

\*\*\* More on importance. \*\*\*

## Approach of This Book

This book shows the methods used to infer networks, using gene networks as the primary example. Both the kinds of data available (e.g. time series data, knockout data), the number of data points, and the quality of the data all influence the choice of algorithms. We will break the methods into components, discuss when to use each variation of each component, and offer both the software and a workflow-driven tool to run a variety of components in a coherent way Understanding these components allows you the reader to create new algorithms by mixing and matching.

# Experimental Data Inputs

## Transcriptome

Transcriptome, or expression data, is the data most often used for gene regulatory network inference. In a typical experiment, an experimentalist will take two groups of genetically identical organisms, where one is a control and one is the test subject. The experimentalist will perturb the test specimens in different ways. For example, one will receive a nutrient and the other won’t. The “expression value” is some measure of the difference of expression between the control and the test specimens.

### Steady-State Data

Steady-state data consists of expression readings taken at a single time point. These are usually taken after the organism has been perturbed in some way and enough time has elapsed for the organism to incorporate the perturbation. Consider for example a group of nitrogen-starved plants. To obtain steady-state data, the experimentalist would add nitrogen to the plants and measure their expression sometime later. That is, once the expression values of the genes have stopped changing as a result of the perturbation. By looking at what happens as the result of many different types of perturbations, networks and functional relationships can be inferred.

Steady-state data is relatively easy to obtain. Most public microarray databases are full of steady-state experiments, making it an inexpensive way to augment other, more expensive forms of data. Databases of steady-state data generally vary by organism, but some popular ones include *The Arabidopsis Information Resource (TAIR)*, an Arabidopsis database available at <http://arabidopsis.org>, *GeneExpDB*, an *E. Coli* database available at <http://chase.ou.edu/oubcf/>, and \*\*\*Add one for yeast\*\*\*. More links to databases are available in Appendix \*\*\*.

Some experiments take genetically identical organisms and perturbs some one way and some in another way. Others change the genetics. For example, knock-out or knock-down experiments fully or partly remove a gene from a specimen. Such experiments allow one to infer the function of the gene that has been fully or partly removed. For the purposes of network inference, a knock-out can tell you whether one gene affects another, directly or indirectly. For example, if knocking out gene g has no effect on gene g’, then we should not draw an edge from g to g’. On the other hand, if knocking out gene g cause g’ to increase, then there is likely to be a direct or indirect repressive effect of g on g’. Frequently, you will find a large set of knock-outs, where each member of the set knocks out a single gene. Occasionally, you will have experiments having pairs of knock-outs.

The converse of a knock-out is an overexpression experiment in which genetic material is inserted that causes some gene g to express itself at a high level either continuously or based on experimental control. Again, if overexpressing g has no effect on g’, then we should not draw an edge from g to g’ . Conversely, if g’ goes up when g is overexpressed, then g has either a direct or indirect inductive effect on g’

Time-series data consists of several microarrays taken over the course of some perturbation (either knock-out, over-expression, or conditional). The plant is perturbed at time 0, and microarrays are taken at different time points. Often, time-series microarray data attempts to capture what happens between the perturbation and the steady-state. [Jesse: the word equilibrium means “death” to most biologists] The dynamics of the system can be modeled by observing how the expression values change over time. If the time points are close enough, we will *assume* that we can model the effects due to single edges. Thus, time series data often allows us to give quantitative values to edges.

So, there are two dimensions in the data: what is perturbed (a condition or genetics) and whether we have a single time point or multiple closely spaced time points.

## Other

\*\*\* RNA-Seq, as suggested by Manny as an up-and-coming new data that may replace microarrays \*\*\* We have so far discussed expression data without saying where it comes from. As of this writing, there are two technologies: …. For the most part, the type of data does not influence our algorithms. Inasmuch as RNA-Seq is higher quality data, it will give us better results.

# Overall Workflow of Inference

Inferring a network entails several different analysis steps. Each step can be done in different ways, and different steps can be taken depending on the available data. For instance, assume that we have an extremely large dataset. Attempting to infer edges in a dataset of more than 1,000 genes may be infeasible because we won’t have the data to distinguish different causal possibilities. For example, if, in a group of experiments, genes g1, g2, and g3 always rise and and fall together, then there will be no way to distinguish a causal link to or from one of these genes from a causal link to or from another. Clustering these genes together to reduce the size of the data set and to eliminate unsupported distinctions among genesis a reasonable first step. Given additional data having complete steady state knock-out data, we may be able to break apart clusters if, say, g1 has a different effect on g4 than g2 does.

Because there are so many ways to build networks, there should be a way to evaluate them. The basic idea is to leave out some data (e.g. the last time point of a time series) and call that test data. The remaining data, called training data, is used to form a variety of networks depending on different methods used. One can then test each network on the test data. The best network suggests the best method. (This is a slight simplification, because it is best to test a method using several random seeds to avoid choosing one method over another for the wrong reason.) Thus, experimentation and testing are required to figure out the best network for a given dataset.

The workflow for inferring gene regulatory networks can be organized into four distinct steps, each of which can be performed in several different ways. The first step clusters the elements of large datasets in order to reduce the size of the inference problem. The second step uses steady-state knock-out data (if available) to infer at least some of the topology of the network. The third step infers a dynamical graph model (when closely spaced time series data is available) whose nodes are either single genes or gene clusters and whose edges are either inductive or repressive edges. The fourth step prunes the dynamical model based on resampling and consensus techniques.

[ Workflow diagram showing the 4 steps above ]